> dynamically modifying a first set of server names fro a first server by adding a first server name to the first set of server names (i.e. set of services), wherein the first server responds to requests directed to the first set of server names (i.e. host system) (e.g. col 2, line 43-45 and col 3, line 37-47);
>
> dynamically modifying a second set of server names for a second server by adding a second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names (i.e. backup) (i.e. col 2, line 43-45 and col 6, line 18-19);
>
> determining that the first server requires reconfiguration (i.e. failure) (e.g. col 8, line 23-37);
>
> dynamically modifying the first set of server names by adding the second server name to the first set of server names (i.e. fail-over) (e.g. col 6, line 54-62).
>
> As per claim 10, Lim teaches the first server is reconfigured in response to a determination that the second server requires fail-over support b the first server (e.g. col 2, line 27-35).
>
> As per claim 11, it is rejected for similar reasons as stated in claim 9. Furthermore, Lim teaches the steps of dynamically modifying the first set of server names by removing the second server name from the first set of server names (i.e. fail-back) (e.g. col 6, line 45-52).

Office Action dated November 8, 2002, pages 2-3.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

Lim does not teach all elements of claims 9-11. Claim 9 reads as follows:

> 9. A method for reconfiguring servers in a distributed data processing system, the method comprising the computer-implemented steps of:
>
> dynamically modifying a first set of server names for a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names;
>
> dynamically modifying a second set of server names for a second server by adding a second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names;
>
> determining that the first server requires reconfiguration; and

B

dynamically modifying the first set of server names by adding the second server name to the first set of server names.

Claim 11 reads as follows:

> 11. A method for reconfiguring servers in a distributed data processing system, the method comprising the computer-implemented steps of:
>
> dynamically modifying a first set of server names for a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names;
>
> dynamically modifying the first set of server names by adding a second server name to the first set of server names;
>
> determining that the first server requires reconfiguration;
>
> dynamically modifying the first set of server names by removing the second server name from the first set of server names; and
>
> dynamically modifying a second set of server names for a second server by adding the second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names.

The Office Action refers to the following portions of *Lim* with respect to the claimed feature of dynamically modifying a first set of server names for a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names:

> Symmetric one-to-one failover is a similar technique, wherein each of the "host" and "backup" systems provide distinct but useful sets of services when both are available, and each is capable of providing the services normally provided by the other.

Lim, column 2, lines 43-45.

> To coordinate the cooperation of the various hosts within the service group a "leader" host is established and assigns to the various hosts responsibility for particular service addresses. The leader identifies resources, such as service addresses (which may, for example, be IP addresses) which are not being attended and causes such resources to be acquired by designated hosts. The leader also dynamically reassigns resources to hosts which have been newly added or restored to the service group, as well as causing the release of resources handled by failed hosts.

Lim, column 3, lines 37-47.

Claims 9 and 11 of the present invention recite that a first set of server names for a first server is modified by adding a first server name to the first set of server names. The first server responds to requests directed to the first set of server names. *Lim* does

13

not teach or suggest this feature, as recited in claims 9 and 11. *Lim* teaches that a leader host assigns responsibility for an identified service address to a specific host rather than that a first server responds to requests directed toward the set of server names. In other words, *Lim* teaches assigning each of a set of service (IP) addresses to different hosts, whereas the presently claimed invention dynamically adds a new server name to a single host so that the server will respond to requests directed to any of the server names in the set of server names. There is no teaching or suggestion in *Lim* to add server names to a set associated with a single server.

Moreover, the differences between the service addresses of *Lim* and the server names of the present application and how they are used are significant. In *Lim*, service addresses are numerical IP addresses (column 5, lines 58-59). IP addresses are used by network routers to route network packets to their destinations and used by network hubs to identify a physical device address (e.g., an Ethernet address) of a network adapter. The host associated with the IP address receives packets sent to that IP address because the IP address maps to a physical device address corresponding to the host's network adapter card.

The rejected claims, however, recite that the ***server*** responds to requests directed toward one of the server names in the set of server names. As can be seen from the specification, the server determines whether an incoming request is directed toward one of its server names, then responds if the received request is directed toward one of the server names for that server, or the server ignores the request if the request is not directed toward one of the server names for the server. Consider Figure 8 from the specification, which is reproduced below:

8

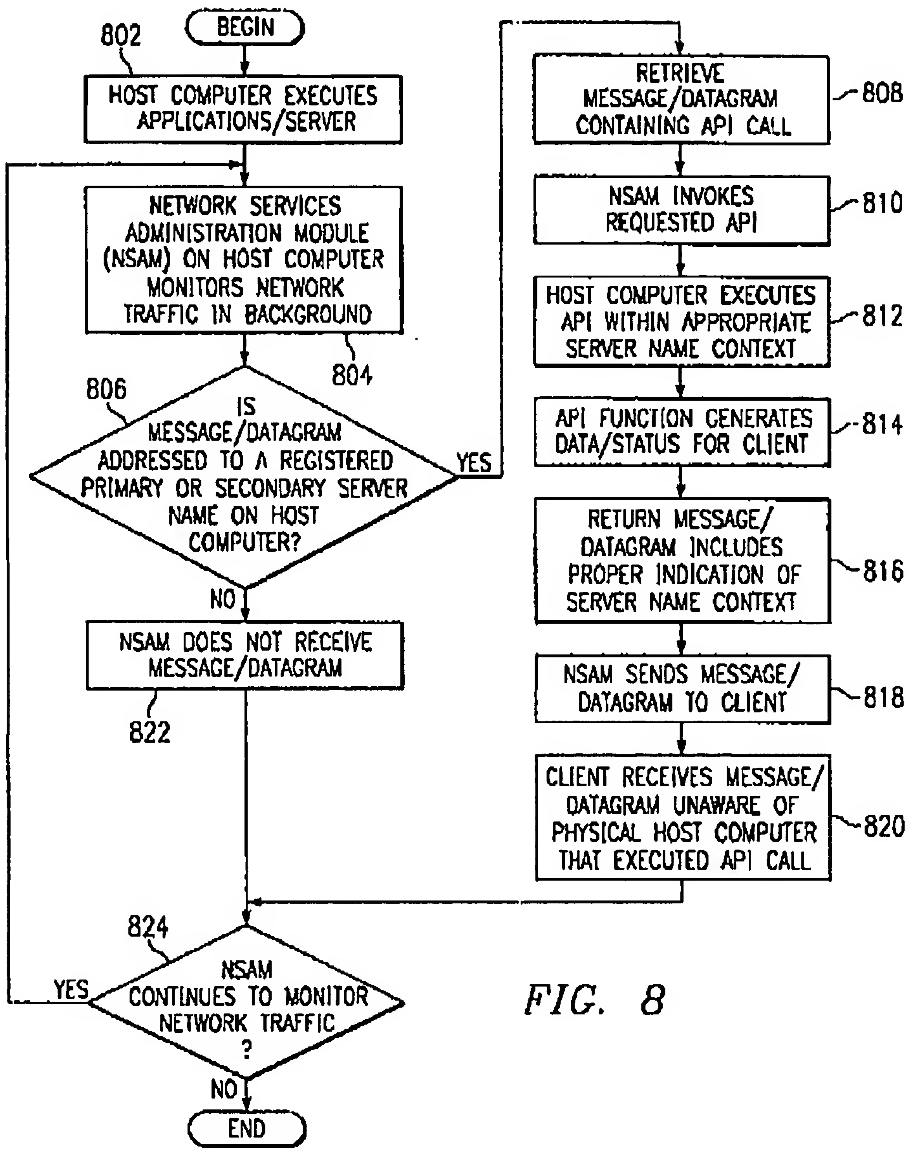

FIG. 8

Specification, Figure 8.

The host monitors network traffic (block 804). If any message or datagram is addressed to a registered server name on the host (block 806: yes), the host responds (blocks 808-820). Otherwise, the message is ignored (block 822).

Thus, unlike in *Lim*, where changing service addresses affects whether messages are received by the ***network adapter***, claims 9-11 recite that the ***server*** responds to those

B

messages that are not only received, but are directed toward one of the server names associated with the server.

Note that this is reflected in Figure 4, in which the NSAM (reference symbol 412) of a host, which is what determines whether to respond or not, sits above the IP (418) and physical network (400) layers, which are the portions of the host's network stack that receive packets based on the physical network addresses and IP addresses. Figure 4 is reproduced below:

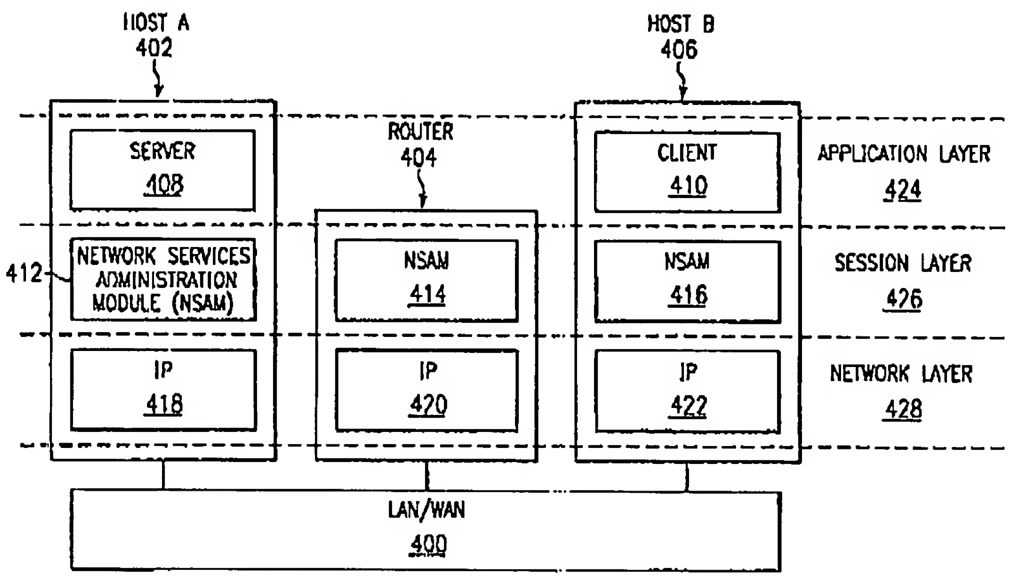

FIG. 4

Specification, Figure 4.

With respect to the claimed feature of dynamically modifying a second set of server names for a second server by adding a second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names, the Office Action refers to the following portions of *Lim*:

> Symmetric one-to-one failover is a similar technique, wherein each of the "host" and "backup" systems provide distinct but useful sets of services when both are available, and each is capable of providing the services normally provided by the other.

Lim, column 2, lines 43-45.

β

> Among other tasks, the leader tries to establish various conditions at equilibrium, including that all, service addresses are being served, that each service address is assigned to only one host, and that each host will serve its preferred service address.

Lim, column 6, lines 18-19.

With respect to the claimed feature of dynamically modifying the first set of server names by adding the second server name to the first set of server names, the Office Action refers to the following portion of *Lim*:

> When a host fails, the leader will detect that the service addresses that were originally served by the failed host are not being actively served, because the control messages will show increasingly long periods since the last time the service addresses were served. The leader then reassigns these unserved addresses to other available hosts within the group. The reassignments may be made randomly, or may be made base on a load-balancing scheme, or through any other suitable method.

Lim, column 6, lines 54-62.

Lim does not teach or suggest dynamically modifying a second set of server names for a second server by adding a second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names and dynamically modifying the first set of server names by adding the second server name to the first set of server names, as recited in claims 9 and 11. *Lim* teaches that one host is assigned a particular service address to service and that the leader host ***reassigns*** an unserved address to another available host within the group of hosts rather than that servers respond to requests directed toward sets of server names and dynamically modifying sets of server names by adding server names to the sets of server names, as recited in the rejected claims. These features allow the same server name to exist in multiple sets of server names. Thus, *Lim* actually teaches away from the present invention by teaching the assignment of a single address to only one host. The claimed invention, in contrast, allows for the assignment of the same server name to two servers.

In claim 9, this feature allows both the first server and the second server to respond to requests directed toward the second server name, which exists in both the first set of server names and the second set of server names. In contrast, *Lim* teaches that ***only one*** host is assigned a particular service address.

B

As stated above, Applicants' position is that service addresses and server names are not equivalent. Nonetheless, even if we assume for the sake of argument that the Examiner is correct and that service addresses and server names are interchangeable, *Lim* teaches away from the claimed invention, because *Lim* teaches that a single service address corresponds to ***only one*** host, whereas claim 9 recites adding the same server name to two different sets of server names.

Additionally, with respect to claim 11, the Office Action refers to the following portion of *Lim*:

> The case where a different host has already been assigned the preferred service address of the new host usually arises where a host has previously failed, and is now rejoining the group. In such a situation, the leader will typically request the host currently owning the preferred service address to release it, and will then wait for the release request to complete before assigning the preferred service address to the joining host.

Lim, column 6, lines 45-52.

In claim 11, first and second server names exists in a first set of server names. The second server name is removed from the first set of server names and added to a second set of server names, thereby causing the second server to respond to requests directed toward the second server name and the first server not to respond. In the presently claimed invention, a server responds to requests directed toward a set of server names associated with that server. *Lim* does not teach or suggest the steps of dynamically modifying the first set of server names by removing the second server name from the first set of server names, as recited in claim 11. *Lim* teaches that a host must release a service address prior to the leader host reassigning the service address to another host. Claim 11 does not recite changing a service address or any other IP or network address; claim 11 recites removing a server name from a first set of server names associated with a first server and adding the server name to a second set of server names associated with a second server. There is no teaching or suggestion in *Lim* to dynamically modify a set of server names associated with a server, wherein a server responds to requests directed toward the set of server names as in the presently claimed invention.

Therefore, *Lim* does not teach or suggest the features as recited in independent claims 9 and 11.

B

Since claim 10 depends from claim 9, the same distinctions between *Lim* and the claimed invention exist for this claim. Additionally, claim 10 claims other additional combinations of features not suggested by the reference. Claim 10 recites the method of claim 9 wherein the first server is reconfigured in response to a determination that the second server requires fail-over support by the first server. In the rejection of claim 10 the Office Action refers to the following portion of *Lim*:

> As the need for continuously available computer services has grown, the need for increased scalcability and reliability has also grown. One of the key issues has been to ensure that a service provided by a first computer, normally termed a host, can be provided by another computer, or a backup, in the event the host becomes unavailable. This transfer of services is termed failover, and in current systems is typically handled by software.

Lim, column2, lines 27-35.

Lim does not teach or suggest that the first server is reconfigured (by adding the second server name to the first set of server names) in response to a determination that the second server requires fail-over support by the first server. As stated previously with respect to claim 9, reconfiguring the first server in this way would allow both the first server and the second server to respond to requests directed toward the second server name. *Lim* teaches only that a leader host assigns the responsibility of a particular service address to one host and the leader host can reassign the responsibility to another host in the event that the responsible host becomes unavailable. In other words, although *Lim* teaches a form of failover inasmuch as *Lim* teaches that addresses can be reassigned, *Lim* does not teach reconfiguring a server in the manner recited in claim 9 in response to a determination that another server requires failover support. Consequently, it is respectfully urged that the rejection of claim 10 has been overcome.

Therefore, the rejection of claims 9-11 under 35 U.S.C. § 102 has been overcome.

Furthermore, *Lim* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *Lim* actually teaches away from the presently claimed invention because it teaches assigning each service address in a fixed set to a single host as opposed to dynamically modifying sets of server names associated with each server as in the presently claimed invention. Absent some teaching or incentive to modify *Lim* to dynamically modify sets of server names associated with

Received from < 972 367 2002 > at 2/10/03 3:55:34 PM [Eastern Standard Time]

B

each server, one of ordinary skill in the art would not be led to modify *Lim* to reach the present invention when the reference is examined as a whole. Thus, the presently claimed invention can be reached only through an improper use of hindsight using Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

II. 35 U.S.C. § 103, Obviousness

The examiner has rejected claim 12 under 35 U.S.C. § 103(a) as being unpatentable over *Lim*, U.S. Patent Number 5,938,732, in view of *Chrabaszcz*, U.S. Patent Number 6,134,673, 10/17/2000, filed 10/01/1997, "Method for Clustering Software Applications". This rejection is respectfully traversed.

As to claim 12, the office action states:

> As per claim 12, Lim does not disclose the limitation of the claim. Chrabaszcz discloses the second server name is removed prior to connecting the second server to a network in the distributed data processing system (i.e. resume operation) (e.g. col 9, line 44-59).). At the time the invention was made, it would have been obvious to a person skill in the art to combine Lim and Chrabaszcz because it is a way to keep data integrity and prevent resource conflicts.

Office Action dated November 8, 2002, page 4.

Since claim 12 depends from claim 11, the same distinctions between *Lim* and the claimed invention exist for this claim. *Lim* does not teach or suggest the features as recited in claims 9 and 11 such as dynamically adding or removing server names from sets of server names, where in a server responds to requests directed to a set of server names. Additionally, claim 12 recites other additional combinations of features not suggested by the references. Claim 12 recites the method of claim 11 wherein the second server name is removed from the first set of server names prior to connecting the second server to a network in the distributed data processing system. In the rejection of claim 12, the Office Action refers to the following portion of *Chrabaszcz*:

> When the first server 102 resumes normal operations, the cluster aware application is loaded into the first server 102. However, it is in a pause mode as a result of a built-in feature of cluster aware applications. Prior to allowing itself to execute, the cluster aware application checks for conflicts. The cluster aware application checks the database 110 with respect to the object which represents

B

> the cluster aware application and notes that server 102 is the primary server for the cluster aware application, but is not the host server. It further notes that the second server 104 is assigned as the host server. Therefore, the cluster aware application is aware that it is a primary server coming out of failure. The clustered application that has been loaded into the primary server memory will not be executed until it verifies that the backup server has unloaded the clustered application. The cluster aware application has thus effectively been paused.

Charabaszcz, column 9, lines 44-59.

As stated in the Office Action, *Lim* does not disclose the limitations of claim 12. Additionally, *Chrabaszcz* does not teach or suggest that the second server name is removed prior to connecting the second server to a network in the distributed data processing system. *Chrabaszcz* teaches that a clustered application that has been loaded into primary server memory will not be executed until it is verified that the backup server has unloaded the clustered application. This is different from the claimed feature in which the second server name is removed from the first set of server names prior to connecting the second server to a network in the distributed data processing system. Consequently, it is respectfully urged that the rejection of claim 12 has been overcome.

Therefore, the rejection of claim 12 under 35 U.S.C. § 103 has been overcome.

β

III. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 10 February 2003

Respectfully submitted,

Michael R. Nichols

Michael R. Nichols
Reg. No. 46,959
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

Received from < 972 367 2002 > at 2/10/03 3:55:34 PM [Eastern Standard Time]

ß

APPENIX OF CLAIMS

9. A method for reconfiguring servers in a distributed data processing system, the method comprising the computer-implemented steps of:

dynamically modifying a first set of server names for a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names;

dynamically modifying a second set of server names for a second server by adding a second server namo to the second set of server names, wherein the second server responds to requests directed to the second set of server names;

determining that the first server requires reconfiguration; and

dynamically modifying the first set of server names by adding the second server name to the first set of server names.

10. The method of claim 9 wherein the first server is reconfigured in response to a determination that the second server requires fail-over support by the first server.

11. A method for reconfiguring servers in a distributed data processing system, the method comprising the computer-implemented steps of:

dynamically modifying a first set of server names for a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names;

dynamically modifying the first set of server names by adding a second server name to the first set of server names;

determining that the first server requires reconfiguration;

dynamically modifying the first set of server names by removing the second server namo from the first set of server names; and

dynamically modifying a second set of server names for a second server by adding the second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names.

Received from < 972 367 2002 > at 2/10/03 3:55:34 PM [Eastern Standard Time]

K

12. The method of claim 11 wherein the second server name is removed prior to connecting the second server to a network in the distributed data processing system.

23. (New) A method for reconfiguring servers in a distributed data processing system, the method comprising the computer-implemented steps of:

dynamically modifying a first set of server names stored on a first server by adding a first server name to the first set of server names, wherein the first server responds to requests directed to the first set of server names;

dynamically modifying a second set of server names stored on a second server by adding a second server name to the second set of server names, wherein the second server responds to requests directed to the second set of server names;

determining that the first server requires reconfiguration; and

dynamically modifying the first set of server names by adding the second server name to the first set of server names.

B